UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/567,851	10/06/2006	Jouni Kytomaa	39700-638N01US/NC40070US	7339

64046 7590 11/24/2009

MINTZ, LEVIN, COHN, FERRIS, GLOVSKY AND POPEO, P.C
ONE FINANCIAL CENTER
BOSTON, MA 02111

EXAMINER	
MITCHELL, DANIEL D	

ART UNIT	PAPER NUMBER
2477	

MAIL DATE	DELIVERY MODE
11/24/2009	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/567,851	KYTOMAA ET AL.	
	Examiner	Art Unit	
	DANIEL MITCHELL	2477	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *30 July 2009*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-17; 35-46* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-17; 35-46* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *10 February 2006* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☒ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

Response to Amendment

1. Applicant's amendment filed on 7/30/2009 has been entered. Claims 1, 35, and 46 have been amended. Claims 18-34 are canceled. Claims 1-17, and 35-46 are still pending in this application, with claims 1, 35, and 46 being independent.

Response to Arguments

2. Applicant's arguments with respect to claims 1-17 and 35-46 have been considered but are moot in view of the new ground(s) of rejection.

Claim Rejections - 35 USC § 103

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

4. Claims 1-17 and 35-46 are rejected under 35 U.S.C. 103(a) as being unpatentable over Ayers (US Patent No. 7,426,209 B2), in view of Vinnakota et al. (US Patent No. 6,789,056 B2), hereinafter referred as Vinnakota in further view of Tuck, III et al. (US Patent No. 6,738,378 B2), hereinafter referred as Tuck.

Regarding claim 1, Ayres teaches a method comprising:

allocating each received packet to at least one arrival queue **(fig. 3, col. 4 lines**

32-48 teaches a data sorter which receives packets and places them in one of a plurality of queues);

placing each packet in the allocated queue **(col. 4 lines 32-48 teaches allocating a packet to the queue)**;

scheduling, by a scheduler coupled to the at least one arrival queue, packets from the arrival queue to at least one processor **(col. 4 lines 32-48 teaches scheduling a packet from a queue to a processor).**

wherein the scheduler includes a first quantity N of inputs each corresponding to the at least one arrival queue, the scheduler further including a second quantity M of outputs, wherein the second quantity M is less than or equal to the first quantity N **(fig. 3, col. 4 lines 32-48 teaches a scheduler with a plurality of inputs and a plurality of outputs where the number of inputs is less than the number of outputs).**

However Ayers does not expressly disclose receiving the packet at the transfer queues; responsive to receipt of an interrupt, allocating the packet from said transfer queue to one of a plurality of processor queues; placing the packet in the allocated processor queue; scheduling packets from the processor queues to be processed.

Vinnakota teaches in col. 4 lines 58 to col. 5 line 2 teaches the interface 114 of the processor contains a plurality of transfer queues. Vinnakota further teaches col. 5 lines 15-66 being responsive to an interrupt and allocating a packet from a transfer queue (element 114) to one of a plurality of processor

queues (element 116). Vinnakota further teaches col. 5 line 55 to col. 6 line 10 scheduling a packet to be processed based on priority.

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the teachings of Ayers to include generating an interrupt signal when a packet is received at a transfer queue. One would be motivated as such in order minimize the processing performed by a packet processor col. 3 lines 11-23.

However Ayers and Vinnakota do not expressly disclose wherein if the queues are full, dropping the packet.

Tuck discloses in col. 6 lines 47-65 that packets are added to queues if the queue is not full, otherwise packets are discarded.

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the teachings of Ayers and Vinnakota to include discarding packets. One would be motivated as such in order to allow the CPU to process data according to the priority during the presence of an overload condition col. 2 lines 20-25.

Regarding claim 2, Ayer teaches wherein packets are received at an input to a plurality of device (fig. 3, col. 4 lines 32-48 teach receiving a packet at an input to a plurality of devices (input queues)).

Regarding claim 3, Ayers teaches in fig. 3, col. 4 lines 32-48 wherein at least one device has a plurality of arrival queues (fig. 3 lines 32-38 teaches a plurality of arrival queues).

Regarding claim 4, Ayers teaches in fig. 3, col. 4 lines 32-48 wherein each arrival queue is associated with a traffic class, each packet being allocated to the at least one queue in accordance with the traffic class of each packet (fig. 3, col. 4 lines 32-48 teaches packets are classified based on the data type, which is interpreted as the traffic class).

Regarding claim 5, Ayers teaches wherein the traffic class is priority information embedded in the each packet (fig. 4 teaches the priority (class) of the packet is embedded in the packet).

Regarding claim 6, Ayers, Vinnakota, and Tuck teach a method as the parent claim.

However Ayers does not expressly disclose the plurality of transfer queues.

Vinnakota teaches a packet processor as the primary reference (col. 3 lines 11-23). Vinnakota teaches in col. 4 lines 26-41 the device with a plurality of transfer queues.

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the teachings of Ayers to include a plurality of transfer queues. One would be motivated as such in order minimize the processing performed by a packet processor col. 3 lines 11-23.

Regarding claim 7, Ayers teaches wherein the number of transfer queues is less than the number of arrival queues (fig. 3, col. 4 lines 32-48 teaches the interface to the scheduler includes less outputs corresponding to the transfer queues than the inputs corresponding to the arrival queues.

Regarding claim 8, Ayers teaches in fig. 3, col. 4 lines 32-48 wherein the scheduling of packets from the arrival queue is dependent upon the traffic profile (col. 4, lines 32-48 teaches wherein the scheduler schedules packets based on traffic arrival time).

Regarding claim 9, Ayer , Vinnakota, and Tuck teach a method as the parent claim.

However Ayers does not expressly disclose wherein the transfer queue comprises a device level transfer queue and a processor level transfer queue, wherein the device level transfer queue receives packets from the arrival queue, and the processor level transfer queue receives packets from the device level transfer queue.

Vinnakota teaches in col. 5 lines 3-14 a transfer queue with a device level transfer queue and a processor level transfer queue. The RAM 122 serves as the device level transfer queues and the queues 140-150 serves as the processor level transfer queues. Vinnakota further teaches in col. 5 lines 3-14 the processor level queues receives packets from the device level queue.

See similar motivation as claim 1.

Regarding claim10, Ayer, Vinnakota, and Tuck teach a method as the parent claim.

However Ayers does not expressly disclose wherein packets are transferred to the processor level transfer queue from the device level transfer queue whenever there is space in the processor level transfer queue.

Vinnakota teaches a packet processor as the primary reference (col. 3 lines 11-23). Vinnakota teaches in col. 5 lines 3-14 that when available packets are transferred from the device level queue to the processor level transfer queue.

See similar motivation as claim 1.

Regarding claim 11, Ayers , Vinnakota, and Tuck teach a method as the parent claim.

However Ayers does not expressly disclose wherein packets are never dropped from the transfer queue.

Vinnakota teaches a packet processor as the primary reference (col. 3 lines 11-23). Vinnakota teaches in col. 3 lines 10-23 that packets are only stored in the transfer queue of device 114. Packets are not dropped from the transfer queue.

See similar motivation as claim 1.

Regarding claim 12, Ayer, Vinnakota, and Tuck teach a method as the parent claim. Ayer further teaches a processor in fig. 3, col. 4 lines 32-48.

However Ayers does not expressly disclose wherein the processor queues are associated with different priorities.

Vinnakota teaches a packet processor as the primary reference (col. 3 lines 11-23). Vinnakota teaches in col. 5 lines 53-66 a processor memory 116 which includes a plurality of queues that are associated with different priorities.

See similar motivation as claim 1.

Regarding claim 13, Ayer , Vinnakota, and Tuck teach a method as the parent claim. Ayer further teaches utilizing queues based on a priority (fig. 3 col. 4 lines 32-48).

However Ayer and Vinnakota do not expressly disclose wherein the highest priority queue has the lowest drop probability and the lowest latency.

Tuck teaches in col. 6 lines 42-65 where the high priority queue has the lowest drop probability since packets are discarded from the low priority queue first and Tuck further teaches the high priority queue has the lowest latency since the high priority queue is serviced prior to servicing the other queues.

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the teachings of Ayers and Vinnakota to include discarding packets. One would be motivated as such in order to allow the CPU to process data according to the priority during the presence of an overload condition col. 2 lines 20-25.

Regarding claim 14, Ayer , Vinnakota, and Tuck teach a method as the parent claim.

However Ayer does not expressly disclose wherein responsive to receipt of the interrupt, a packet is removed from a transfer queue and classified.

Vinnakota teaches a packet processor as the primary reference (col. 3 lines 11-23). Vinnakota teaches in col. 5 lines 38-66 a interrupt is created and as a result the packet is removed from the transfer queues and placed in the processor queues based on the priority classification given to the packet.

See similar motivation as claim 1.

Regarding claim 15, Ayer , Vinnakota, and Tuck teach a method as the parent claim.

However Ayers does not expressly disclose wherein the classification is based on a determination of priority.

Vinnakota teaches a packet processor as the primary reference (col. 3 lines 11-23). Vinnakota teaches in col. 5 lines 38-66 15 wherein the classification of packets into the processor queues is based on the priority given to the packet

See similar motivation as claim 1.

Regarding claim 16, Ayer, Vinnakota, and Tuck teach a method as the parent claim.

However Ayers does not expressly disclose wherein the packet is allocated to a processor queue in accordance with a classification of the packet.

Vinnakota teaches a packet processor as the primary reference (col. 3 lines 11-23). Vinnakota teaches in col. 5 lines 38-66 15 wherein the alloaction of packets into the processor queues is based on the priority given to the packet

See similar motivation as claim 1.

Regarding claim 17, Ayer, Vinnakota, and Tuck teach a method as the parent claim.

However the combination does not expressly disclose wherein the packet is placed in the allocated processor queue if said queue is not full, otherwise the packet is dropped.

Tuck teaches in col. 6 lines 42-65 where packets are allocated to a queue unless the queues become full. In that instance, packets are dropped.

See similar motivation as claim 1.

Regarding claim 35, Ayres teaches a apparatus comprising: processor configured to allocate each received packet to at least one arrival queue **(fig. 3, col. 4 lines 32-48 teaches a data sorter which receives packets and places them in one of a plurality of queues; col. 5 line 42-45 teaches the processor)**;

placing each packet in the allocated queue **(col. 4 lines 32-48 teaches allocating a packet to the queue);**

scheduling, by a scheduler coupled to the at least one arrival queue, packets from the arrival queue to at least one processor **(col. 4 lines 32-48 teaches scheduling a packet from a queue to a processor).**

wherein the scheduler includes a first quantity N of inputs each corresponding to the at least one arrival queue, the scheduler further including a second quantity M of outputs, wherein the second quantity M is less than or equal to the first quantity N **(fig. 3, col. 4 lines 32-48 teaches a scheduler with a plurality of inputs and a plurality of outputs where the number of inputs is less than the number of outputs).**

However Ayers does not expressly disclose receiving the packet at the transfer queues; responsive to receipt of an interrupt, allocating the packet from

said transfer queue to one of a plurality of processor queues; placing the packet in the allocated processor queue; scheduling packets from the processor queues to be processed.

Vinnakota teaches in col. 4 lines 58 to col. 5 line 2 teaches the interface 114 of the processor contains a plurality of transfer queues. Vinnakota further teaches col. 5 lines 15-66 being responsive to an interrupt and allocating a packet from a transfer queue (element 114) to one of a plurality of processor queues (element 116). Vinnakota further teaches col. 5 line 55 to col. 6 line 10 scheduling a packet to be processed based on priority.

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the teachings of Ayers to include generating an interrupt signal when a packet is received at a transfer queue. One would be motivated as such in order minimize the processing performed by a packet processor col. 3 lines 11-23.

However Ayers and Vinnakota do not expressly disclose wherein if the queues are full, dropping the packet.

Tuck discloses in col. 6 lines 47-65 that packets are added to queues if the queue is not full, otherwise packets are discarded.

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the teachings of Ayers and Vinnakota to include discarding packets. One would be motivated as such in order to allow

the CPU to process data according to the priority during the presence of an overload condition col. 2 lines 20-25.

Regarding claim 36, Ayer teaches wherein packets are received at an input which includes a plurality of arrival queues (fig. 3, col. 4 lines 32-48 teaches a plurality of arrival queues at an input).

Regarding claim 37, Ayers teaches in fig. 3, col. 4 lines 32-48 wherein each arrival queue is associated with a traffic class, each packet being allocated to the at least one queue in accordance with the traffic class of each packet (fig. 3, col. 4 lines 32-48 teaches packets are classified based on the data type, which is interpreted as the traffic class).

Regarding claim 38, Ayers, Vinnakota, and Tuck teach a method as the parent claim.

However Ayers does not expressly disclose the plurality of transfer queues.

Vinnakota teaches a packet processor as the primary reference (col. 3 lines 11-23). Vinnakota teaches in col. 4 lines 26-41 the device with a plurality of transfer queues.

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the teachings of Ayers to include generating

an interrupt signal when a packet is received at a transfer queue. One would be motivated as such in order minimize the processing performed by a packet processor col. 3 lines 11-23.

Regarding claim 39, Ayer , Vinnakota, and Tuck teach a apparatus as the parent claim.

However Ayers does not expressly disclose wherein the transfer queue comprises a device level transfer queue and a processor level transfer queue, wherein the device level transfer queue receives packets from the arrival queue, and the processor level transfer queue receives packets from the device level transfer queue.

Vinnakota teaches in col. 5 lines 3-14 a transfer queue with a device level transfer queue and a processor level transfer queue. The RAM 122 serves as the device level transfer queues and the queues 140-150 serves as the processor level transfer queues. Vinnakota further teaches in col. 5 lines 3-14 the processor level queues receives packets from the device level queue.

See similar motivation as claim 35.

Regarding claim 40, Ayer, Vinnakota, and Tuck teach an apparatus as the parent claim.

However Ayers does not expressly disclose wherein packets are transferred to the processor level transfer queue from the device level transfer queue whenever there is space in the processor level transfer queue.

Vinnakota teaches a packet processor as the primary reference (col. 3 lines 11-23). Vinnakota teaches in col. 5 lines 3-14 that when available packets are transferred from the device level queue to the processor level transfer queue

See similar motivation as claim 35.

Regarding claim 41, Ayers , Vinnakota, and Tuck teach an apparatus as the parent claim.

However Ayers does not expressly disclose wherein packets are never dropped from the transfer queue.

Vinnakota teaches a packet processor as the primary reference (col. 3 lines 11-23). Vinnakota teaches in col. 3 lines 10-23 that packets are only stored in the transfer queue of device 114. Packets are not dropped from the transfer queue.

See similar motivation as claim 35.

Regarding claim 42, Ayer, Vinnakota, and Tuck teach an apparatus as the parent claim. Ayer further teaches a processor in fig. 3, col. 4 lines 32-48.

However Ayers does not expressly disclose wherein the processor queues are associated with different priorities.

Vinnakota teaches a packet processor as the primary reference (col. 3 lines 11-23). Vinnakota teaches in col. 5 lines 53-66 a processor memory 116 which includes a plurality of queues that are associated with different priorities.

See similar motivation as claim 35.

Regarding claim 43, Ayer , Vinnakota, and Tuck teach an apparatus as the parent claim.

However Ayer does not expressly disclose wherein responsive to receipt of the interrupt, a packet is removed from a transfer queue and classified.

Vinnakota teaches a packet processor as the primary reference (col. 3 lines 11-23). Vinnakota teaches in col. 5 lines 38-66 a interrupt is created and as a result the packet is removed from the transfer queues and placed in the processor queues based on the priority classification given to the packet.

See similar motivation as claim 35.

Regarding claim 44, Ayer, Vinnakota, and Tuck teach an apparatus as the parent claim.

However Ayers do not expressly disclose wherein the packet is allocated to a processor queue in accordance with a classification of the packet.

Vinnakota teaches a packet processor as the primary reference (col. 3 lines 11-23). Vinnakota teaches in col. 5 lines 38-66 15 wherein the allocation of packets into the processor queues is based on the priority given to the packet

See similar motivation as claim 35.

Regarding claim 45, Ayer, Vinnakota, and Tuck teach an apparatus as the parent claim.

However the Ayer and Vinnakota do not expressly disclose wherein the packet is placed in the allocated processor queue if said queue is not full, otherwise the packet is dropped.

Tuck teaches in col. 6 lines 42-65 where packets are allocated to a queue unless the queues become full. In that instance, packets are dropped.

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the teachings of Ayers and Vinnakota to include discarding packets. One would be motivated as such in order to allow the CPU to process data according to the priority during the presence of an overload condition col. 2 lines 20-25.

Regarding claim 46, a computer-readable storage medium encoded with instructions that, when executed on a computer, perform a process **(col. 6 lines30-35 teaches a memory for storing a program; col. 7 lines 33-36 teaches a processor for performing a process)** , the process comprising: processor configured to allocate each received packet to at least one arrival queue **(fig. 3, col. 4 lines 32-48 teaches a data sorter which receives packets**

and places them in one of a plurality of queues; col. 5 line 42-45 teaches the processor);

placing each packet in the allocated queue **(col. 4 lines 32-48 teaches allocating a packet to the queue)**;

scheduling, by a scheduler coupled to the at least one arrival queue, packets from the arrival queue to at least one processor **(col. 4 lines 32-48 teaches scheduling a packet from a queue to a processor).**

wherein the scheduler includes a first quantity N of inputs each corresponding to the at least one arrival queue, the scheduler further including a second quantity M of outputs, wherein the second quantity M is less than or equal to the first quantity N **(fig. 3, col. 4 lines 32-48 teaches a scheduler with a plurality of inputs and a plurality of outputs where the number of inputs is less than the number of outputs.**

However Ayers does not expressly disclose receiving the packet at the transfer queues; responsive to receipt of an interrupt, allocating the packet from said transfer queue to one of a plurality of processor queues; placing the packet in the allocated processor queue; scheduling packets from the processor queues to be processed.

Vinnakota teaches in col. 4 lines 58 to col. 5 line 2 teaches the interface 114 of the processor contains a plurality of transfer queues. Vinnakota further teaches col. 5 lines 15-66 being responsive to an interrupt and allocating a packet from a transfer queue (element 114) to one of a plurality of processor

queues (element 116). Vinnakota further teaches col. 5 line 55 to col. 6 line 10 scheduling a packet to be processed based on priority.

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the teachings of Ayers to include generating an interrupt signal when a packet is received at a transfer queue. One would be motivated as such in order minimize the processing performed by a packet processor col. 3 lines 11-23.

However Ayers and Vinnakota do not expressly disclose wherein if the queues are full, dropping the packet.

Tuck discloses in col. 6 lines 47-65 that packets are added to queues if the queue is not full, otherwise packets are discarded.

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the teachings of Ayers and Vinnakota to include discarding packets. One would be motivated as such in order to allow the CPU to process data according to the priority during the presence of an overload condition col. 2 lines 20-25.

Conclusion

5. Any response to this action should be **faxed** to (571) 173-8300 or **mailed** to:

Commissioner of Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Hand delivered responses should be brought to:

Customer Service Window
Randolph Building
401 Dulany Street
Alexandria, VA 22314

6. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to DANIEL MITCHELL whose telephone number is (571)270-5307. The examiner can normally be reached on Monday - Friday 8:00 am - 5:00 pm EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Chirag G. Shah can be reached on 571-272-3144. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/D. M./
Examiner, Art Unit 2477

/Chirag G Shah/
Supervisory Patent Examiner, Art Unit 2477